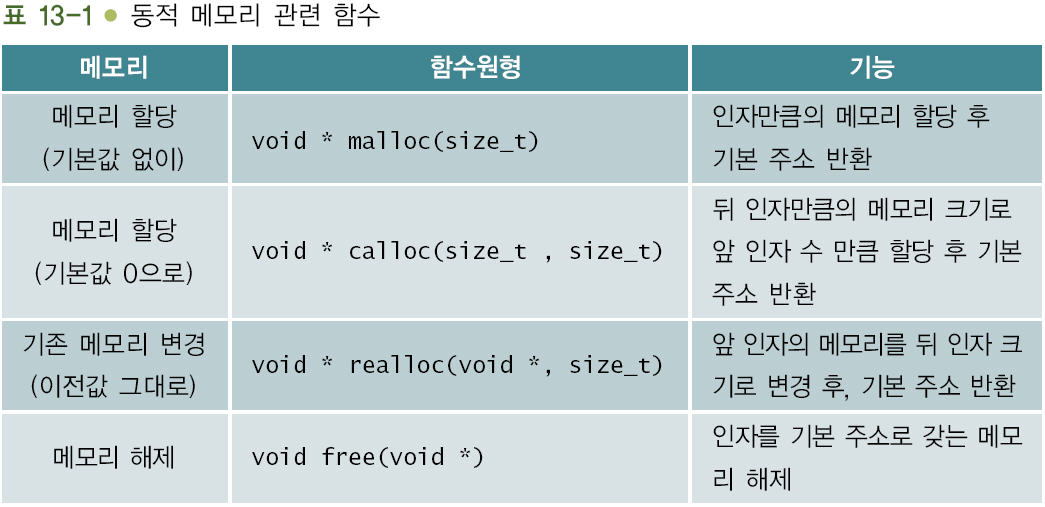
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2018\_2\_C++ \_05 | 학번 : |  | 이름 : |  |

1. C언어 동적 메모리 할당
2. 동적 메모리 할당

* **malloc(), calloc(), realloc() 3가지**
* **반환값의 유형은 모든 자료형의 포인터로 이용할 수 있도록 void \* => 형변환 필요**
* **<stdlib.h>에 함수 원형 정의**



**===== malloc() 활용 예**

#include <stdlib.h>

#include <stdio.h>

int main(void) { //동적 메모리에 여러 성적을 저장하여 합과 평균을 구하여 출력

int \*ary = NULL;

int i = 0, n = 0, sum= 0;

printf("입력할 점수의 개수를 입력>> ");

scanf\_s("%d", &n);

//입력한 개수만큼의 메모리 할당

if ( (ary = (int \*) malloc( sizeof(int)\*n )) == NULL ) {

printf("메모리 할당에 문제가 있습니다.");

exit(1);}

printf("%d개의 점수 입력>> ", n);

for (i = 0; i < n; i++) {

scanf\_s("%d", (ary + i) ); //할당된 공간의 주소값 계산

sum += \*(ary + i); //sum += ary[i];, 할당된 공간 참조

}

printf("합: %d 평균: %.1f\n", sum, (double)sum/n);

free(ary);

return 0;}

**====== calloc() 활용 예**

#include <stdlib.h>

#include <stdio.h>

int main(void) {

int \*ary = NULL;

int i = 0;

if ( (ary = (int \*) calloc( 3, sizeof(int) )) == NULL ){

printf("메모리 할당에 문제가 있습니다.\n");

exit(EXIT\_FAILURE);

}

//메모리 할당 후 초기화 과정 생략 시 0으로 저장

for (i = 0; i < 3; i++)

printf("ary[%d] = %d\n", i, \*(ary + i));

free(ary);

return 0;

}

**======= realloc 활용 예**

#include <stdlib.h>

#include <stdio.h>

int main(void) {

int \*reary, \*cary;

int i = 0;

if ((cary = (int \*)calloc(3, sizeof(int))) == NULL) {

printf("메모리 할당에 문제가 있습니다.\n");

exit(EXIT\_FAILURE);

}

if ((reary = (int \*)realloc(cary, 4 \* sizeof(int))) == NULL) {

printf("메모리 할당이 문제가 있습니다.\n");

exit(EXIT\_FAILURE);

}

printf("정수데이터를 입력하세요 : \n");

for (i = 0; i < 3; i++)

scanf\_s("%d", (cary + i));

printf("\n정수데이터를 출력합니다 : \n");

for (i = 0; i < 4; i++)

printf("reary[%d] = %d\n", i, \*(reary + i));

free(reary);

return 0;

}

1. **프로그램 과제** : 다음과 같은 구조체를 사용하여 제시된 결과처럼 실행되는 프로그램을 작성하시오. - 입력된 차량 개수만큼 저장할 수 있는 메모리 공간을 할당 받아 처리한다.

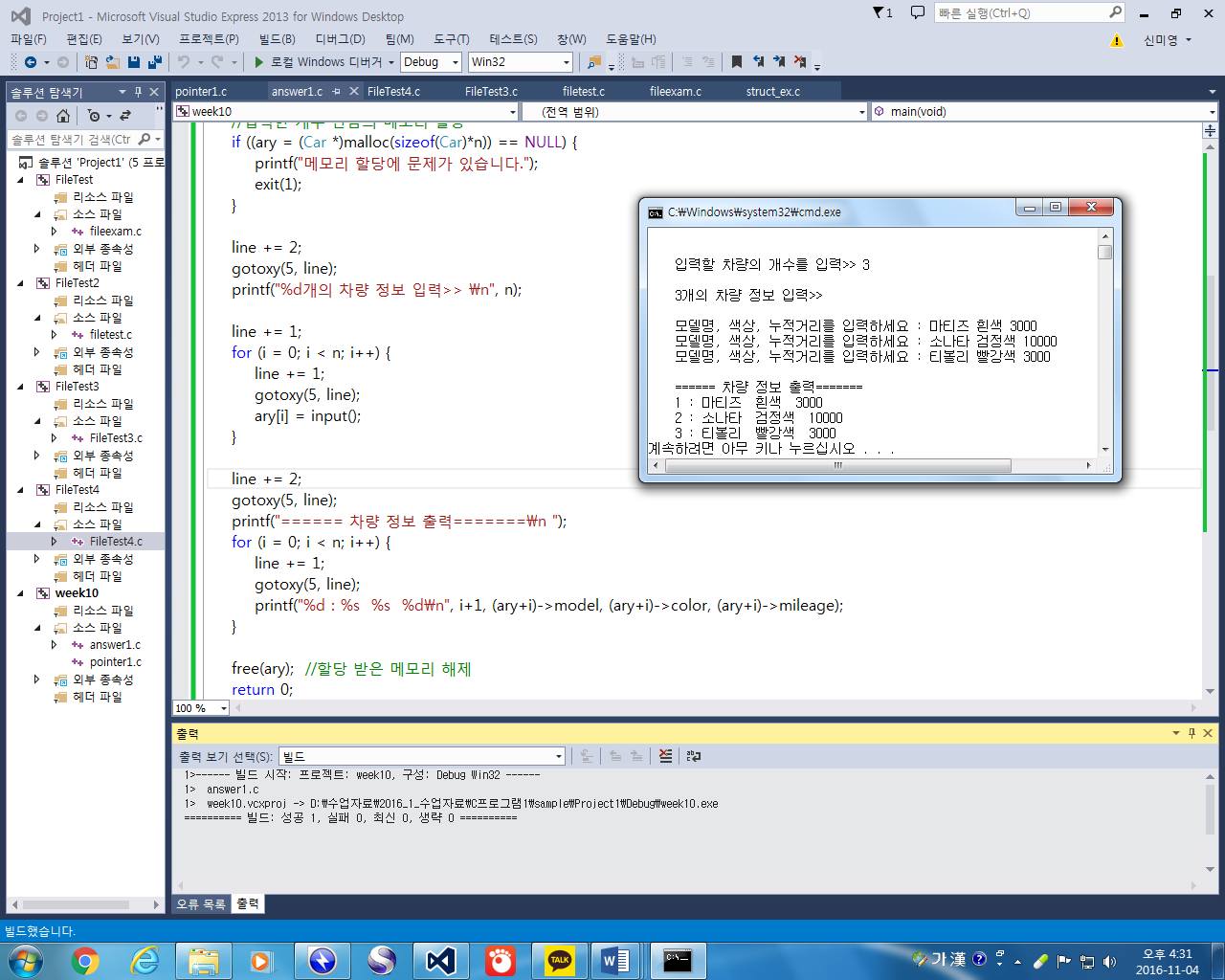
**typedef struct{**

**char model[20]; //모델명**

**char color[10]; //색상**

**int mileage; //누적 거리**

**}Car;**



|  |
| --- |
| **[프로그램 소스]**  #include<stdio.h>  #include<stdlib.h>  typedef struct {  char model[20]; //모델명  char color[20]; //색상  int mileage; //누적 거리  }Car;  void main() {  int size=0;  Car \*c;  char che;  printf("입력할 차량의 개수를 입력 >>");  scanf\_s("%d",&size);  printf("\n");  c = (Car\*)malloc(sizeof(Car\*)\*size);  printf("%d 개의 차량 정보 입력\n\n",size);  for (int i = 0; i < size; i++) {  printf("모델명, 색상, 누적거리를 입력하세요.");  scanf\_s("%s",c[i].model,20);  scanf\_s("%s", c[i].color,10);  scanf\_s("%d", &c[i].mileage);  }  che = getchar();  printf("\n======차량 정보 출력=====\n");  for (int i = 0; i < size; i++) {  printf("%d : %s\t%s\t%d\n",(i+1),c[i].model,c[i].color,c[i].mileage);  }  che = getchar();  } |
| **[실행 결과]** |

* 내용 점검

1. 교재 4장 연습문제

|  |  |
| --- | --- |
| 번호 | 문제풀이 |
| 7 | 2. 기본 생성자 Cube() 가 4번 호출 된다. |
| 8 | 기본생성자  기본생성자  기본생성자  소멸자  소멸자  소멸자 |
| 10 | Cube \*p = new Cube[4]; 는 객체 배열 로 생성을 했기에 소멸시엔 delete[] p 와 같이 배열 표기를 해서 소멸시에도 배열임을 알려줘야 하나 그냥 delete p 로 되어 있다 |
| 12 | 3번 static 함수를 포함하여 맴버함수 내에서만 다루어지는 객체 자신에 대한 포인터 이다.  non-static 에서만 사용 가능 하다. |
| 15 | 1.메모리 누수 X 2. 메모리 누수 0 -> int \*p = new int ;int \*q =p; delete q 로 하면 q의 메모리만 제거 되고 \*p = new int 로 동적으로 받은 메모리는 반환이 안된다. |
| 18 | 3. string 클래스가 좋기는 하지만 c++ 표준이 아니므로 가급적 사용자지 않는 것이 좋다.  ->string 클래스는 c++ 표준 라이브러리 에서 제공하는 클래스 |
| 19 | string a = "name";  char ch = a.at(2);  if (a == "name")  cout << "same"<<endl;  a.append("~~");  a.insert(1, "y"); |

1. 교재 5장 연습문제

|  |  |
| --- | --- |
| 번호 | 문제풀이 |
| 1 | 4.묵시에 의한 호출 |
| 2 | 1.값에 의한 호출 |
| 3 | 주소에 의한 호출 |
| 4 | 1.0(같다) 2. X(다르다) |
| 5 | 5, 25 |
| 6 | 1,4,9 |
| 7 | Char r & = c; |
| 8 | q = 'a'; |
| 9 | 1,2,3,4 |
| 10 | 3. |
| 11 | void copy(int &a, int &b) {a = b;} |
| 12 | 100, 값에 값은 대입이 불가능 하기 떄문에. |

* 실행 결과 확인

1. 기본 타입 변수에 대한 참조

#include <iostream>

using namespace std;

int main() {

cout << "i" << '\t' << "n" << '\t' << "refn" << endl;

int i = 1;

int n = 2;

int &refn = n; // 참조 변수 refn 선언. refn은 n에 대한 별명

n = 4;

refn++; // refn=5, n=5

cout << i << '\t' << n << '\t' << refn << endl;

refn = i; // refn=1, n=1

refn++; // refn=2, n=2

cout << i << '\t' << n << '\t' << refn << endl;

int \*p = &refn; // p는 refn의 주소를 가짐. 실제 p는 n의 주소를 가짐

\*p = 20; // refn=20, n=20

cout << i << '\t' << n << '\t' << refn << endl;

}

1. 객체에 대한 참조

#include <iostream>

using namespace std;

class Circle {

int radius;

public:

Circle() { radius = 1; }

Circle(int radius) { this->radius = radius; }

void setRadius(int radius) { this->radius = radius; }

double getArea() { return 3.14\*radius\*radius; }

};

int main() {

Circle circle;

Circle &refc = circle;

refc.setRadius(10);

cout << refc.getArea() << " " << circle.getArea(); // 두 호출은 동일 객체에 대한 호출

}

1. 참조에 의한 호출

#include <iostream>

using namespace std;

class Circle {

int radius;

public:

Circle() { radius = 1; }

Circle(int radius) { this->radius = radius; }

void setRadius(int radius) { this->radius = radius; }

double getArea() { return 3.14\*radius\*radius; }

};

**void readRadius(Circle &c)** {

int r;

cout << "정수 값으로 반지름을 입력하세요>>";

cin >> r; // 반지름 값 입력

c.setRadius(r); // 반지름 설정

}

int main() {

Circle donut;

readRadius(donut);

cout << "donut의 면적 = " << donut.getArea() << endl;

}

1. 참조 리턴

#include <iostream>

using namespace std;

// 배열 s의 index 원소 공간에 대한 참조 리턴하는 함수

char& find(char s[], int index) {

return s[index]; // 참조 리턴

}

int main() {

char name[] = "Mike";

cout << name << endl;

find(name, 0) = 'S'; // name[0]='S'로 변경

cout << name << endl;

char& ref = find(name, 2);

ref = 't'; // name = "Site"

cout << name << endl;

}

* 프로그램 과제

1. 교재 212p : 문제5번

|  |
| --- |
| **[프로그램 소스]**  #include<iostream>  #include<string>  #include<cstring>  #include<cstdlib>  #include<ctime>  using namespace std;  int main() {  string str;  while (1) {  cout << "아래에 한 줄을 입력하세요.(exit 를 입력하면 종료 합니다.)" << endl;  getline(cin, str);  if (str == "exit" || str == "EXIT") {  break;  }  else {  srand((unsigned)time(0));  int x = rand() %str.length(), y = rand() % str.length();  str.replace(x,1,str.substr(y,1));  cout << str<<endl;  }  }  } |
| **[실행 결과]** |

1. 교재 213p : 문제9번

|  |
| --- |
| **[프로그램 소스]**  #include<iostream>  #include<string>  #include<cstring>  using namespace std;  class person {  string name;  string tel;  public:  person();  string getname() { return name; }  string gettel() { return tel; }  void set(string name, string tel);  };  person::person() {  name = "Noname";  tel = "000-0000-0000";  }  void person::set(string name, string tel) {  this->name = name;  this->tel = tel;  }  int main() {  string n, t;  person p[3];  int index = -1;  for (int i = 0; i < 3; i++) {  cout <<"사람 " << (i + 1)<<">> " ;  cin >> n >> t;  p[i].set(n, t);  }  cout << "모든 사람의 이름은 ";  for (int i = 0; i < 3; i++) {  cout << p[i].getname()<<" \t";  }  while (1) {  cout << "\n전화번호 검색을합니다. 이름을 입력하세요. >> ";  cin >> n;  for (int i = 0; i < 3; i++) {  if (n == p[i].getname() )  index = i;  }  if (index > -1) {  cout << "\n전화 번호는 "<<p[index].gettel();  break;  }  else {  cout << "\n없는 이름 입니다. 다시 입력 하세요.";  continue;  }    }  char che = getchar();  che = getchar();  } |
| **[실행 결과]** |

1. 교재 216p : 문제12번

|  |
| --- |
| **[프로그램 소스]**  #include<iostream>  #include<string>  #include<cstring>  using namespace std;  class Circle {  int radius;  string name;  public:  void setCircle(string name, int radius);  double getArea();  string getName();  };  class CircleManager {  Circle \*p;  int size;  public:  CircleManager(int size);  ~CircleManager();  void searchByname();  void searchByArea();  };  void Circle::setCircle(string name, int radius) {  this->name = name;  this->radius = radius;  }  double Circle::getArea() {  return (3.14\*radius\*radius);  }  string Circle::getName(){  return name;  }  CircleManager::CircleManager(int size) {  string n;  int r;  this->size = size;  p = new Circle[size];  for (int i = 0; i < size; i++) {  cout << "원 " << i + 1 << "의 이름과 반지름 >> ";  cin >> n >> r;  p[i].setCircle(n, r);  }  }  CircleManager::~CircleManager() {  delete[] p;  }  void CircleManager::searchByname() {  string n;  cout << "\n검색하고자 하는 원의 이름 >> ";  cin >> n;  for (int i = 0; i < size; i++) {  if (n == p[i].getName()) {  cout << n<<"의 면적은 "<< p[i].getArea();  }else {}  }  }  void CircleManager::searchByArea() {  int r;  cout << "\n최소 면적을 정수로 입력하세요 >> ";  cin >> r;  for (int i = 0; i < size; i++) {  if (p[i].getArea() > r) {  cout << p[i].getName() << "의 면적은" << p[i].getArea() << ", ";  }else{ }  }  }  int main() {  int s;  string n;  cout << "원의 개수 >> ";  cin >> s;  CircleManager cm(s);  cm.searchByname();  cm.searchByArea();  char che = getchar();  che = getchar();  } |
| **[실행 결과]** |

1. 교재 270p : 문제3번

|  |
| --- |
| **[프로그램 소스]**  #include<iostream>  #include<string>  #include<cstring>  using namespace std;  void combine(string text1, string text2, string &text3) {  text1.append(" ");  text1.append(text2);  text3 = text1;  }  int main() {  string text1("I love you"), text2("very much"),text3;  combine(text1, text2, text3);  cout << text3;  char che = getchar();  } |
| **[실행 결과]** |

1. 교재 270p : 문제4번

|  |
| --- |
| **[프로그램 소스]**  #include<iostream>  #include<string>  #include<cstring>  using namespace std;  bool bigger(int a, int b, int &big);  int main() {  int a = 3, b = 4, big=0;  if (bigger(a, b, big))  cout << "true" << endl;  else  cout << "false " << big << "입니다" << endl;  char che = getchar();  }  bool bigger(int a, int b, int &big) {  if (a == b) {  return true;  }  else {  if (a > b) {  big = a;  }else {  big = b;  }  return false;  }  } |
| **[실행 결과]** |

1. 교재 271p : 문제6번

|  |
| --- |
| **[프로그램 소스]**  #include<iostream>  #include<string>  #include<cstring>  using namespace std;  char& find(char a[], char c, bool& success);  int main() {  char s[] = "Mike";  bool b = false;  char& loc = find(s, 'M', b);  if (b == false) {  cout << "M을 발견할 수 없다" << endl;  return 0;  }  loc = 'm';  cout << s << endl;  char che = getchar();  }  char& find(char a[], char c, bool& success) {  for (int i = 0; i < sizeof(a)/sizeof(a[0]); i++) {  if (a[i] == c) {  success = true;  return a[i];  }else {  success = false;  }  }  return \*a;  } |
| **[실행 결과]** |